Amendments to the Claims

Claim 1 (currently amended): A method for programmatically enforcing referential integrity constraints among associations between class instances, comprising steps of:

determining, when evaluating a request to set an association end to reflect an association from an instance of a first class to an instance of a second class, whether the association end has a single multiplicity or a many multiplicity;

setting the requested association end; and

programmatically modifying an inverse association end of the association to reflect an inverse association from the instance of the second class to the instance of the first class, after disconnecting the inverse association end from an existing instance, if any;

wherein an ordering of the setting step and the programmatically modifying step depends on an outcome of the determining step.

Claim 2 (canceled)

Claim 3 (currently amended): The method according to Claim [[2]] 1, wherein the ordering of the setting and the programmatically modifying steps for a particular association end where the outcome is ~~that has a~~ the single multiplicity further ~~comprise steps of:~~ comprises

~~disconnecting the inverse association end from an existing instance, if any;~~

~~performing the programmatically modifying step after performing the disconnecting step;~~

~~and~~

performing the setting step after performing the ~~disconnecting~~ programmatically modifying step.

Claim 4 (currently amended): The method according to Claim [[2]] 1, wherein the ordering of the setting and the programmatically modifying steps for a particular association end where the outcome is the ~~that has a~~ many multiplicity further ~~comprise steps of:~~ comprises

performing the setting step before performing the programmatically modifying step;

~~disconnecting the inverse association end from an existing instance, if any, after performing the setting step; and~~

~~performing the programmatically modifying step after performing the setting step.~~

Claim 5 (original): The method according to Claim 1, further comprising steps of:

determining whether the association end or the inverse association end is a primary end of the association; and

serializing only the primary end of the association during a serialization operation.

Claim 6 (original): The method according to Claim 1, wherein the method is provided as link helper objects.

Claim 7 (currently amended): A computer program product for programmatically enforcing referential integrity constraints among associations between class instances, wherein the

computer program product is embodied on one or more computer readable media and comprises:

computer-readable program code means for determining, when evaluating a request to set an association end to reflect an association from an instance of a first class to an instance of a second class, whether the association end has a single multiplicity or a many multiplicity;

computer-readable program code means for setting the requested association end; and

computer-readable program code means for programmatically modifying an inverse association end of the association to reflect an inverse association from the instance of the second class to the instance of the first class, after disconnecting the inverse association end from an existing instance, if any;

wherein an ordering of operating the computer-readable program code means for setting and the computer-readable program code means for programmatically modifying depends on an outcome of the computer-readable program code means for determining.

Claim 8 (canceled)

Claim 9 (currently amended): The computer program product according to Claim [[8]] 7, wherein the ordering of operating the computer-readable program code means for setting and the computer-readable program code means for programmatically modifying for a particular association end where the outcome is the ~~that has a~~ single multiplicity further comprises ~~comprise:~~

~~computer-readable program code means for disconnecting the inverse association end~~

~~from an existing instance, if any;~~

~~computer-readable program code means for operating the computer-readable program code means for programmatically modifying after operating the computer-readable program code means for disconnecting; and~~

computer-readable program code means for operating the computer-readable program code means for setting after operating the computer-readable program code means for ~~disconnecting~~ programmatically modifying; and

wherein the ordering of operating the computer-readable program code means for setting and the computer-readable program code means for programmatically modifying for a particular association end ~~that has a~~ where the outcome is the many multiplicity further comprises ~~comprise:~~

computer-readable program code means for operating the computer-readable program code means for ~~performing the~~ setting before operating the computer-readable program code means for programmatically modifying;

~~computer-readable program code means for disconnecting the inverse association end from an existing instance, if any, after operation of the computer-readable program code means for setting; and~~

~~computer-readable program code means for operating the computer-readable program code means for programmatically modifying after operating the computer-readable program code means for setting.~~

Claim 10 (original): The computer program product according to Claim 7, further comprising:

computer-readable program code means for determining whether the association end or the inverse association end is a primary end of the association; and

computer-readable program code means for serializing only the primary end of the association during a serialization operation.

Claim 11 (currently amended): A system for programmatically enforcing referential integrity constraints among associations between class instances, comprising:

means for determining, when evaluating a request to set an association end to reflect an association from an instance of a first class to an instance of a second class, whether the association end has a single multiplicity or a many multiplicity;

means for setting the requested association end; and

means for programmatically modifying an inverse association end of the association to reflect an inverse association from the instance of the second class to the instance of the first class, after disconnecting the inverse association end from an existing instance, if any;

wherein an ordering of operating the means for setting and the means for programmatically modifying depends on an outcome of the means for determining.

Claim 12 (canceled)

Claim 13 (currently amended): The system according to Claim [[13]] 11, wherein the ordering of

operating the means for setting and the means for programmatically modifying for a particular association end ~~that has a~~ where the outcome is the single multiplicity further comprises ~~comprise:~~

~~means for disconnecting the inverse association end from an existing instance, if any;~~

~~means for operating the means for programmatically modifying after operating the means for disconnecting; and~~

~~means~~ means for operating the means for setting after operating the means for ~~disconnecting~~ programmatically modifying; and

wherein the ordering of operating the means for setting and the means for programmatically modifying for a particular association end ~~that has a~~ where the outcome is the many multiplicity further comprises ~~comprise:~~

~~means~~ means for operating the means for ~~performing the~~ setting before operating the means for programmatically modifying;

~~means for disconnecting the inverse association end from an existing instance, if any, after operation of the means for setting; and~~

~~means for operating the means for programmatically modifying after operating the means for setting.~~

Claim 14 (original): The system according to Claim 11, further comprising:

means for determining whether the association end or the inverse association end is a primary end of the association; and

means for serializing only the primary end of the association during a serialization operation.